## AMENDMENT OF THE SPECIFICATION

Applicant respectfully requests that the following paragraphs replace corresponding paragraphs of the specification in the "DETAILED DESCRIPTION OF EMBODIMENTS" section. The changes to these paragraphs correct typos and do not add new matter:

**[0015]** Server **130** may include a computer system such as an IBM eServer™ having one or more processors, or threads of processors, executing software and/or one or more state machines coupled with data storage devices such as random access memory (RAM), read only memory (ROM), flash memory, compact disc drives, hard drives, and the like. Software executing on server **130** may be adapted to receive, process, and respond to messages. The software may also be adapted to gather and return data from server **130** or other computer systems coupled with WAN/LAN **150** that have compatible messaging software and/or data storage. Server **1[[1]]30** may include an acceptor **132**, queues **134**, a queue manager **139**, a dispatcher **140**, a thread pool **142**, and process logic **144**.

**[0021]** Thread pool **142** may include available, working, and dirty threads associated with one or more processors to process messages stored in working queue **136**. The threads may store a committed reply for the message in outbound queue **137** prior to removing the message from working queue **136**, advantageously maintaining persistence via local queue manager **139** from receipt of the message until a committed reply is stored and persisted in outbound queue **137**. For example, when a system failure such as the shutdown of server **130** may occur after receipt of a message by inbound queue **135** and before a reply is generated, the persisted messages can be restored to working queue **136[[7]]** and processed upon restarting server **130**. In addition, when a system failure occurs after a committed reply is stored in outbound queue **137** but before the reply is transmitted to the requestor, the reply may be transmitted upon start up of server **130**. In further embodiments, when this persistence is accomplished via a middleware application, the applications executing via the middleware application may be designed for efficiency without regard for persistence since persistence is inherent to processing messages.

[0025] WAN/LAN **150** is a network connection to couple workstation **110** and server **130** with user computers such as hosts **160**, **170**, and **180** to transmit messages between the computers. In some embodiments, WAN/LAN **150** may include a network in an office coupled via Ethernet, optical media like ~~OptiConnect~~OPTICONNECT™, or the like. In some embodiments, WAN/LAN **150** also couples with the Internet via a cable modem, a direct subscriber line (DSL), a T1 line, a T3 line, or the like. In further embodiments, WAN/LAN **150** may include a network of temporary connections such as connections via a telephone system. In other embodiments, one or more user computers like host **170** may couple with workstation **110** directly.

[0032] Thread designator **240** may couple with searcher **232** to assign request **210** to a thread of thread pool **250** when the thread is available to process request **210**. More specifically, when dispatcher **230** copies request **210** to working queue **245**, thread designator **240** may check thread pool **250** for available threads **252**. The next available thread may be assigned to request **210** to process request **210**. When no threads are available to process request **210**, dispatcher **23[[1]]0** may wait for a thread to become available. In other embodiments, dispatcher **230** may not copy request **210** to working queue **245** until at least one thread is available or a dirty thread is being cleaned. In several embodiments, an entry in working queue **245** may not be available for request **210** unless at least one thread is available or available to be cleaned. In further embodiments, any thread may be able to process any message such as request **210**.

[0038] Upon accessing rules **256**, the thread assigned to a message may access process logic **260** to process the message. Process logic **260** may include presentation logic manager **262**, presentation logic **264**, business logic manager **266**, business logic **268**, data logic **270**, and database **272**. Presentation logic manager **262** may manage one or more presentation logic programs like presentation logic **264** to interface with a display associated with apparatus **200**. For instance, when request **210** is associated with a presentation logic program, the thread assigned to request **210** may couple with process logic **260** to display a message or graphic on the display associated with apparatus **200**.

**[0046]** Middleware **320** may coordinate processing of messages and persistence of messages or their corresponding replies. In particular, middleware **320** includes queues **324** and queue manager **326**. Queues **324** and queue manager **326** may involve logic similar to queues **220**, **245[[22]]**, and **280**, and queue manager **224**, from FIG 2. For instance, queue manager **326** may persist the contents of queues **324**.